

На правах рукописи

Пупырев Сергей Николаевич

МОДЕЛИ, АЛГОРИТМЫ И ПРОГРАММНЫЙ КОМПЛЕКС  
ВИЗУАЛИЗАЦИИ СЛОЖНЫХ СЕТЕЙ

05.13.18. – математическое моделирование, численные методы и комплексы программ

Автореферат

диссертации на соискание ученой степени  
кандидата физико-математических наук

Екатеринбург — 2010

Работа выполнена в Государственном образовательном учреждении высшего профессионального образования «Уральский государственный университет им. А.М. Горького» на кафедре алгебры и дискретной математики.

Научный руководитель: доктор физико-математических наук,  
профессор  
Баранский Виталий Анатольевич

Официальные оппоненты: доктор технических наук,  
профессор  
Гайдамакин Николай Александрович

кандидат физико-математических наук  
Волканин Леонид Сергеевич

Ведущая организация: Институт математики и механики Уральского  
отделения РАН

Защита состоится \_\_\_\_ декабря 2010 в \_\_\_\_ ч. на заседании Диссертационного совета Д 212.286.10 при ГОУ ВПО «Уральский государственный университет им. А.М. Горького» по адресу: 620000, г. Екатеринбург, пр. Ленина 51, комн. 248.

С диссертацией можно ознакомиться в Научной библиотеке ГОУ ВПО «Уральский государственный университет им. А.М. Горького».

Автореферат разослан \_\_\_\_ ноября 2010.

Ученый секретарь Диссертационного совета  
доктор физико-математических наук, профессор

Пименов В.Г.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы.** Графы – это простая, мощная, элегантная абстракция, применяемая во многих областях науки и техники. Графы позволяют моделировать произвольные системы, представимые в виде набора объектов и связей между ними. За последние десятилетия популярность графов значительно возросла. Объясняется это их универсализмом и независимостью от предметной области, что позволяет обрабатывать и анализировать системы произвольной сложности. Информационный взрыв, вызванный развитием Всемирной Паутины, и развитие компьютерных технологий сделали доступным большое количество информации. Сегодня теория графов используется в биологии для анализа взаимодействий между протеинами, в радиоэлектронике – при проектировании печатных схем, в химии – для описания кинетики сложных реакций, в экономике – при оптимизации маршрутов и грузоперевозок, в социологии – для изучения связей и закономерностей в обществе и т.п. Графы являются центральным инструментом при разработке программного обеспечения, структурировании бизнес-процессов, проектировании баз данных и используются во многих областях математики и компьютерных наук.

Считается, что 90% информации человек получает посредством зрения и только 10% через остальные органы чувств. Естественно, что проблема *визуализации* графовой информации приобрела первостепенную важность. Задача визуализации состоит в создании изображения, позволяющего анализировать структуру графа и выявлять его характеристики. В данной диссертации мы концентрируемся на визуализации *«сложных сетей»*. Под сложной сетью понимается система, состоящая из реальных объектов и связей между ними. Другими словами, сложная сеть – это граф, построенный на основе реальных данных. Для сложных сетей характерна безмасштабная топология [1], и они, как правило, наделены дополнительной семантикой, которая обычно выражается набором атрибутов, приписанных вершинам и ребрам графа. Примерами сложных сетей могут служить сети страниц WWW, социальные сети, графы соавторства ученых, бизнес-отношения, сети взаимодействия протеинов и т.д.

Самые ранние изображения графов, дошедшие до наших дней, относятся к XIII веку [9]. Хорошо известны изображения генеалогических деревьев, которые появились, по-видимому, в начале XV века. В переписке Эйлера и Лейбница в XVIII веке обсуждается использование рисунков для решения графовых задач. Позже J.V. Listing, A. Cayley и другие известные математики формулируют и решают задачи рисования деревьев и графов. К началу XX века рисование графов выходит за пределы математики: изображения социальных отношений являются центральной темой в работах Морено в 1932 [7], который считается родоначальником анализа социальных сетей.

Новейшая история теории рисования графов начинается с 80-х годов и связана с появлением автоматических алгоритмов визуализации. В научном обществе формируется группа ученых целенаправленно занимающаяся визуализацией графов. Теория рисования графов выделяется в отдельную область математики. Ежегодно проводится крупная международная конференция (Symposia on Graph Drawing), на которой обсуждаются новейшие алгоритмы и актуальные задачи в области рисования графов. Большое количество конференций имеют выделенные секции по данной тематике (например, Symposium on Discrete Algorithms, Symposium on Computational Geometry, International Conference on Information Visualization и др.). Существует несколько международных журналов, публикующих результаты исследований по визуализации графов. В настоящее время разработано огромное количество программ, библиотек и специализированных пакетов для рисования графов (крупнейшие – graphviz от AT&T Labs, MSAGL от Microsoft, yFiles от yWorks). В нашей стране задачами визуализации информации и рисованием графов занимаются, к примеру, в Новосибирском государственном университете, Уральском государственном университете, Институте математики и механики УрО РАН.

Каковы успехи теории рисования графов на сегодняшний день? Решенными задачами на сегодня можно считать рисование графов с «хорошей» простой структурой – деревьев, сеток, планарных графов, ациклических (в случае ориентированных графов). Для малых и средних графов с количеством вершин, не превосходящим несколько десятков, существуют

эффективные алгоритмы, строящие качественные понятные изображения. Для сложных сетей, имеющих более сложную структуру, существующие методы работают неудовлетворительно. К настоящему времени не была решена задача масштабирования: не были разработаны *методы для визуализации больших реальных графов*. С ростом количества информации актуальными стали вопросы визуализации изменяющихся во времени графов. Не существовало удобных функциональных *систем для интерактивной работы с графовыми данными*.

В силу сказанного выше, проблема визуализации сложных сетей является актуальной и нерешенной задачей. Для полноценного решения задачи нужны модели и алгоритмы рисования графов, а также программные инструменты их визуализации, анализа и обработки.

**Цель данной работы.** Основная цель диссертации заключается в создании интерактивного программного комплекса визуализации, обработки и анализа сложных сетей. Для достижения этой цели решаются следующие задачи.

- Разработать математическую модель сложных сетей, адекватно описывающую реальные данные.
- На основе построенной модели создать алгоритмы визуализации графов, позволяющие решать прикладные задачи анализа сетей.
- Дать теоретические оценки вычислительной сложности разработанных алгоритмов.
- Реализовать алгоритмы в виде программного комплекса.
- Провести экспериментальное тестирование комплекса путем решения прикладных задач, убедившись в его применимости и полезности.

**Методика исследования.** В работе используются методы теории графов и вычислительной геометрии, а также методы дискретной оптимизации и оценки вычислительной сложности алгоритмов. Разработка

программного комплекса основана на методах структурного и объектно-ориентированного программирования с активным использованием шаблонов и моделей.

**Научная новизна.** Все основные результаты диссертационной работы являются новыми. В частности, это алгоритмы визуализации структуры естественных сообществ в графах, метод связывания ребер для поуровневых графов с использованием техники рисования схем метро, а также алгоритмы визуализации динамических сетей. Эффективность и корректность разработанных алгоритмов доказана в ряде теорем.

**Практическая ценность.** Результаты диссертационной работы имеют как теоретическое, так и практическое значение. Автором спроектирован и реализован программный комплекс визуализации графов, который может быть использован для анализа и обработки сложных сетей. Созданный комплекс имеет гибкую архитектуру и возможности для расширения. В него включена функциональность по работе с динамическими графами, интерактивность, богатый пользовательский интерфейс и набор средств для автоматического анализа сетей.

Алгоритмы и модели визуализации графов, разработанные и реализованные автором во время двух трехмесячных стажировок в Microsoft Research, вошли в состав библиотеки MSAGL для автоматического рисования графов. Данная библиотека является основным инструментом визуализации графов в нескольких продуктах, в частности, Microsoft Visual Studio 2010 и Microsoft Code Canvas. Работа автора во время стажировок проводилась в непосредственном сотрудничестве с исследователями Microsoft Research, совместно реализованные алгоритмы уже внедрены в Visual Studio, планируется внедрение в ряд других крупных продуктов.

**Апробация и публикации.** Все основные результаты диссертации отражены в публикациях [11-13],[15-18], список которых приводится в конце автореферата. Публикации [11], [12] и [13] опубликованы в ведущих рецензируемых научных журналах, определенных ВАК. Авторские права на созданные диссертантом совместно с Л. Нахмансоном алгоритмы и программные комплексы оформлены в виде патента [14]. В совместных рабо-

тах [13] и [14] диссертанту принадлежат разработка и реализация алгоритма, а также доказательства всех утверждений, Л. Нахмансону – постановка задачи. Из остальных совместных работ [12] и [18] в диссертацию вошли результаты, полученные лично автором.

Результаты работы докладывались и обсуждались на следующих научных конференциях и семинарах:

1. Всероссийская молодежная школа-конференция «Современные проблемы математики» (Кунгурка). Екатеринбург, 2009 и 2010.
2. Конференция молодых ученых по информационному поиску (RuSSIR). Воронеж, 2010.
3. International Symposium on Graph Drawing (Graph Drawing). Konstanz, Germany, 2010.
4. International Conference on Intelligent Systems Design and Applications (ISDA). Cairo, Egypt, 2010.
5. Научный семинар «Алгебраические системы». УрГУ, Екатеринбург.

Первоначальные результаты работы по укладкам направленных графов и визуализации сложных сетей также представлялись на докладе во время первой трехмесячной научной стажировки автора в Microsoft Research (Рэдмонд, США) в 2009. Работа, описывающая метод связывания ребер для неориентированных графов, была представлена в Microsoft Research в 2010 во время второй трехмесячной научной стажировки. Эта работа была также представлена на международной конференции Symposium on Graph Drawing (Konstanz, Germany) в 2010, которая является крупнейшим ежегодным научным событием для исследователей и разработчиков в области визуализации графов. В рамках конференции между участниками проводится соревнование Graph Drawing Contest, на котором определяются лучшие алгоритмы и методы автоматического рисования графов. В 2010 изображения, построенные автором при помощи метода связывания ребер, были признаны лучшими в номинации «Link Routing».

**Структура работы.** Диссертационная работа состоит из 136 страниц машинописного текста, включающего введение, пять глав и заключение, а также рисунки, таблицы и список литературы, содержащий 115 наименований.

## СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность исследуемой проблемы, сформулирована цель и задачи диссертационной работы, перечислены полученные в диссертации новые результаты, их практическая ценность и описана структура диссертации.

В **главе 1** формулируются классические модели для рисования графов и обсуждаются их основные недостатки для визуализации реальных сложных сетей.

Под «*сложными сетями*» понимаются системы, состоящие из реальных объектов и связей между ними. Сложная сеть моделируется графом, однако этот граф, как правило, имеет определенную структуру и обладает характерными признаками. Такие сети принято называть безмасштабными (*scale-free*), поскольку средняя степень вершины в них не является характерной, т.е. отсутствует характерный масштаб. Для безмасштабной топологии характерно наличие малого числа хабов – вершин наибольшей степени – и большого числа вершин малой степени. Сложные сети имеют хорошо выраженную структуру естественных сообществ: вершины сети разделены по группам, которые слабо связаны между собой, но имеют большую плотность ребер внутри. При этом сложные сети глобально являются разреженными с количеством ребер, пропорциональным количеству вершин  $m = O(n)$ .

В классическом понимании под  $d$ -мерной *укладкой* графа  $G = (V, E)$  понимается отображение вершин графа  $V$  в множество точек и его ребер на жордановы кривые пространства  $R^d$ . Позицию вершины  $v$  будем обозначать через  $p_v$ . Для построения укладки строится специальная модель, в которой вершины и ребра графа соответствуют «реальным» физическим взаимодействующим объектам. Для этой системы вводится функция энер-



гии таким образом, что конфигурации с меньшим уровнем энергии соответствуют лучшим укладкам. При этом задача поиска лучшей укладки графа сводится к поиску минимума энергии системы. В зависимости от способа построения энергии выделяют силовой (*force-directed*) и пружинный (*spring*) алгоритмы рисования графов [2]. В силовой модели вершины графа соответствуют заряженным частицам, между которыми действуют силы притяжения и отталкивания. В общем виде энергия системы выражается следующим образом:

$$U = \sum_{(u,v) \in E} f_a m_{uv} \|p_u - p_v\|^a + \sum_{(u,v): u \neq v} f_r m_u m_v \|p_u - p_v\|^r.$$

Первое слагаемое соответствует энергии притяжения, а второе – энергии отталкивания. Параметры системы  $a, r, f_a, f_r \in R$  и веса вершин и ребер  $m_v, m_{uv} \in R$  определяют степень влияния каждого компонента на итоговую укладку. В пружинной модели ребра графа заменяются пружинами, при растяжении и сжатии которых возникают силы упругости. Энергия системы пружин записывается следующим образом:

$$U = \sum_{(u,v) \in E} \|p_u - p_v\|^2.$$

Задача построения укладки графа в математической модели, основанной на физических аналогиях, сводится к решению оптимизационной задачи поиска укладки с минимальной энергией:

$$\min_{p_1, \dots, p_n \in R^d} U$$

Для решения такой задачи мы используем численные методы оптимизации функций: метод *градиентного спуска* и *мажорирования стресса*. На практике эти модели часто дают похожие результаты. Более того, в диссертации установлено (**теорема 1**), что для любого решения задачи минимизации энергии в пружинной модели существуют параметры силовой модели, для которых это решение также дает минимум.

Классические алгоритмы, основанные на физических аналогиях, имеют ряд существенных недостатков и неприменимы для визуализации сложных сетей. В диссертации разработаны расширения и модификации классических моделей, позволяющие визуализировать графы, построенные на

основе реальных данных. Приведены численные методы для оптимизации соответствующих функций. Сформулируем основные проблемы в области визуализации сложных сетей с использованием классических моделей и алгоритмов.

1. Неравномерное размещение вершин разной степени: вершины с большим количеством смежных ребер оказываются в центре изображения, и плотность вершин к центру изображения заметно возрастает.
2. В современных условиях вершины графа невозможно изобразить точками, поскольку с каждой вершиной ассоциирована некоторая текстовая или графическая информация. Вершины, имеющие размеры, требуется рисовать без пересечений и наложений. Тот же аргумент следует учитывать при рисовании ребер графа: ребра должны изображаться кривыми, не пересекающими вершины.
3. Важным аспектом при визуализации реальных данных является их изменчивость. Сети изменяются во времени, и необходимо визуализировать происходящие изменения. В реальных приложениях часто требуются изображения динамических графов.

Во **второй главе** предлагается алгоритм укладки вершин сложной сети, визуализирующий структуру ее естественных сообществ. Под *сообществом* графа  $G$  мы понимаем подмножество его вершин  $C \subset V$ , в котором вершины тесно связаны друг с другом, и имеют малое количество «внешних» ребер, соединяющих их с остальным графом. Разбиение графа на сообщества – это разбиение множества вершин на непересекающиеся сообщества. Для измерения качества разбиения используется метрика, предложенная в [4], которая называется *модульность (modularity)*:

$$Q = \sum_{k=1}^s \left[ \frac{l_k}{m} - \left( \frac{d_k}{2m} \right)^2 \right], \quad (1)$$

где сумма берется по всем сообществам  $C_1, \dots, C_s$  графа. Через  $l_k$  обозначается количество ребер между вершинами сообщества  $C_k$ , а через  $d_k$  – сумма степеней вершин этого сообщества. Под  $m$ , как обычно, понимается количество ребер в  $G$ . Модульность показывает, насколько «плотность» ребер

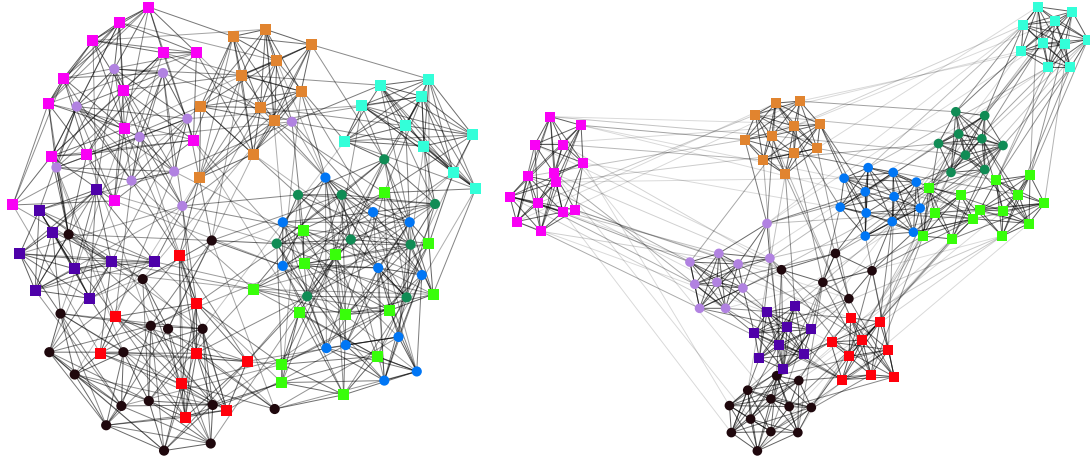
внутри сообществ больше по сравнению со случайным графом, имеющим то же распределение степеней вершин.

Эксперименты показывают (например, в [8]), что большое количество реальных сетей обладают хорошо выраженной структурой сообществ. Более того, можно говорить об *иерархии*, в которой мелкие сообщества оказываются вложенными в более крупные. Каждый уровень иерархии представляет собой набор непересекающихся множеств: верхний уровень содержит единственное сообщество, содержащее все вершины исходного графа, на нижнем уровне количество сообществ совпадает с количеством вершин графа. Сообщества  $l$ -ого уровня будем обозначать  $C_k^l$ , где  $k \in [1.. \text{количество сообществ уровня } l]$ .

Разработанный алгоритм визуализации сообществ состоит из двух этапов. На первом строится иерархическая декомпозиция графа на сообщества. В результате получается дерево сообществ. Второй этап алгоритма – это последовательная (от корня к листьям) укладка сообществ в иерархии. При этом сообщества (и соответствующие вершины исходного графа), имеющие большое количество связей друг с другом, будут расположены рядом друг с другом, несвязанные сообщества будут, напротив, визуально разделены (рис. 1).

Мы строим разбиение графа на сообщества на основе максимизации модульности. Задача поиска разбиения с максимальной модульностью является NP-трудной задачей, поэтому мы используем жадную эвристику. Изначально строится  $n$  сообществ, содержащих по одной вершине графа. После этого на каждом шаге выбирается пара сообществ, объединение которых дает максимальный выигрыш в модульности, и эта пара объединяется. Процедура заканчивает работу, когда объединение любой пары не дает увеличения модульности. Далее мы строим фактор-граф, вершинами которого являются найденные сообщества. Два сообщества в таком графе соединены ребром тогда и только тогда, когда соединены ребром какие-либо вершины из этих сообществ. Рекурсивно вызвав процедуру объединения, мы получаем искомую иерархию сообществ.

На втором шаге мы последовательно обходим все узлы построенного дерева сообществ, начиная с корня, и строим укладку. При обработке корня



(a) Классический силовой алгоритм. (b) Новый алгоритм визуализации сообществ.

Рис. 1. Граф, построенный на основе расписания футбольных игр в американском колледже в 2000 году (115 вершин, 613 ребер). Вершины графа соответствуют командам, ребра – играм между ними. Встречи происходят чаще между командами из одного дивизиона, поэтому можно считать дивизионы естественными сообществами в графе. Наш алгоритм с большой точностью нашел все сообщества графа.

иерархии мы строим фактор-граф  $G'$ . Его вершинами являются сообщества второго уровня  $C_1^2, C_2^2, \dots, C_d^2$ , и каждая пара сообществ, имеющая смежные вершины, соединена ребром весом  $\frac{w(C_i, C_j)}{|C_i||C_j|}$ , где  $w(C_i, C_j)$  – количество ребер между вершинами сообществ  $C_i$  и  $C_j$ . Далее строится укладка графа  $G'$  с использованием силовой модели, в которой учитываются веса ребер. В результате мы получаем укладку, в которой «связанные» сообщества будут расположены рядом друг с другом.

Последовательно спускаясь по дереву, мы укладываем каждый узел-сообщество. Рассмотрим обработку узла  $C_k^l$  с  $l \geq 2$ , т.е. сообщество не менее второго уровня. Соответствующий граф  $G'$  содержит вершины  $C_{i_1}^{l+1}, C_{i_2}^{l+1}, \dots, C_{i_s}^{l+1}$ , т.е. всех детей  $C_k^l$ . Вес ребра между вершинами  $C_i^{l+1}$  и  $C_j^{l+1}$  пропорционален количеству ребер между ними:  $\frac{w(C_i^{l+1}, C_j^{l+1})}{|C_i^{l+1}||C_j^{l+1}|}$ . В силовую модель укладки добавляется дополнительная гравитационная сила, которая притягивает каждую вершину  $C_i^{l+1}$  к позиции вершины  $C_k^l$ , найденной на предыдущем шаге обхода дерева. Таким образом, мы приближенно сохраняем позиции сообществ верхнего уровня относительно друг друга и

одновременно оптимизируем укладку вершин каждого отдельного сообщества. Обход иерархии сообществ реализуется при помощи обхода в ширину.

**Теорема 2.** *Вычислительная сложность алгоритма визуализации сообществ составляет  $O(n^3)$ .*

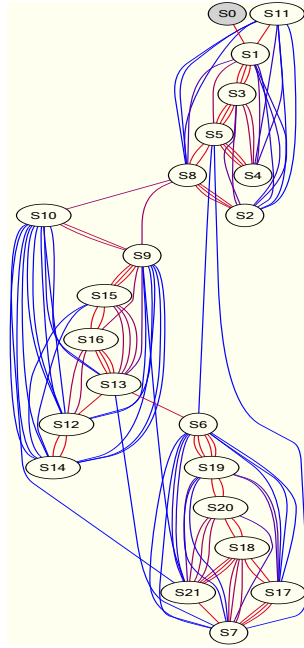
В отличие от произвольных графов, сложные сети, построенные на основе реальных данных, обладают рядом особенностей, позволяющих создавать более эффективные алгоритмы. В частности, реальные графы являются разреженными, т.е. количество ребер в графе пропорционально количеству вершин:  $m = O(n)$ . Другое известное свойство – иерархия сообществ реальных сетей, полученная методом оптимизации модульности, имеет высоту порядка  $\log n$  [3]. Будем называть граф *реальным*, если он имеет оба этих свойства.

**Теорема 3.** *Алгоритм визуализации сообществ имеет вычислительную сложность  $O(n^2 \log n)$  для реальных графов.*

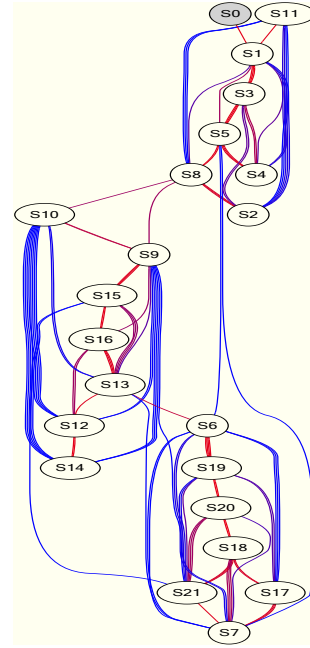
Мы провели ряд экспериментов применения нашего алгоритма для визуализации нескольких реальных графов, которые часто используются при тестировании алгоритмов анализа данных и кластеризации сложных сетей. Для всех рассмотренных реальных графов мы нашли разбиение на сообщества с большим значением модульности, что подтверждает гипотезу о существовании естественных сообществ в реальных сетях. Качество получаемых изображений было протестировано на нескольких графах с заранее известной структурой: оно оказалось лучше, чем полученные популярными алгоритмами (рис. 1).

**Глава 3** посвящена рисованию ориентированных графов. Основным результатом главы является разработка модификации классической поуровневой схемы Сугиямы для случая плотных графов.

В поуровневой схеме рисования графов [10] строится разбиение вершин графа  $G = (V, E)$  на подмножества  $L_1, \dots, L_h$  так, что для всех дуг графа  $e = (u, v)$ , где  $u \in L_i$  и  $v \in L_j$ , выполняется  $i < j$ . После построения разбиения вершин исходный граф преобразуется к *строго поуровневому* графу  $G^p$ . Для этого каждая дуга  $e = (u, v)$  с  $u \in L_i$  и  $v \in L_j$ , проходящая через несколько уровней, заменяется на последовательность вершин



(a) Стандартная укладка



(b) Связывание ребер

Рис. 2. Диаграмма состояний приложения *notepad.exe*.

$u = d_i, d_{i+1}, \dots, d_j = v$  и ребер  $(d_i, d_{i+1}), (d_{i+1}, d_{i+2}), \dots, (d_{j-1}, d_j)$ . Вершины  $d_k$  для  $i < k < j$  называются *фиктивными*, а вершины  $d_i$  и  $d_j$  – *реальными*. Для фиктивных и реальных вершин строится порядок, при котором количество пересечений ребер графа  $G^p$  минимально, и для каждой вершины определяется ее горизонтальная координата.

Стандартный способ для рисования ребер в такой схеме – это проведение кусочнолинейных ломаных (или сплайнов) через соответствующие вершины поуровневого графа. Однако если количество ребер в графе велико, то даже в оптимальной укладке ребра пересекаются слишком часто и визуально сложно определить, какие вершины соединены ребром, а какие нет. Мы предлагаем рисовать некоторые ребра вместе друг с другом, тем самым экономя пространство для рисования и делая изображение проще для понимания структуры графа (рис. 2). Такой метод называется *связыванием ребер* (*edge bundling*).

На вход алгоритму связывания ребер подается поуровневый граф  $G^p$ , разбиение на уровни  $L$  и укладка, т.е. координаты его вершин. Алгоритм состоит из трех шагов.

1. Сгруппировать ребра графа  $G^p$  в набор связок (*bundles*), где под связ-

кой понимается множество одноуровневых ребер. Группировка ребер происходит на основе минимизации *длины укладки* графа (*ink*). Под длиной укладки мы понимаем суммарную длину ребер нарисованного графа, при этом совпадающие ребра считаются один раз.

Задачу поиска укладки поуровневого графа с минимальной длиной мы решаем с помощью «жадной» эвристики, поскольку общая задача минимизации длины укладки является NP-трудной. Изначально для каждого ребра графа  $G^p$  мы создаем связку, содержащую только это ребро. Далее мы находим пару связок, объединение которых максимально уменьшает длину укладки, и производим это объединение. Процедура объединения связок производится до тех пор, пока длина укладки графа уменьшается.

2. Укладка графа с минимальной длиной, полученная после предыдущего этапа связывания, может иметь дуги с большим количеством изломов и поворотов. Для разрешения этой проблемы мы корректируем координаты некоторых фиктивных вершин, «выпрямляя» дуги и делая их более вертикальными и плавными.
3. После объединения ребер в связки изображение графа заметно упрощается, но при этом появляется элемент «двусмысленности», т.е. не всегда возможно отследить путь дуги от начальной вершины к конечной. На этом этапе алгоритма мы «расширяем» каждую связку таким образом, что входящие в ее состав ребра рисуются индивидуальными параллельными отрезками. При расширении связок возникает вопрос об относительном порядке ребер в пределах каждой связки. Естественно использовать тот порядок, при котором количество пересечений дуг исходного графа  $G$  минимально. Соответствующая задача называется минимизацией пересечений линий в схемах метро (*metro-line crossing minimization problem*, [6]). Все пары дуг графа  $G$  можно разделить на два класса: те, которые можно нарисовать (т.е. указать порядок в пределах связки) без пересечений, и «неизбежные» – те, которые пересекаются при любом указанном порядке.

**Теорема 6.** *Оптимальный порядок дуг графа  $G$  в пределах каждой*

связки содержит только неизбежные пересечения.

**Теорема 7.** Сложность алгоритма вычисления порядка с минимальным количеством пересечений дуг графа  $G$  составляет  $O(|E^p| + |E| \log |E| + L_p)$ , где  $|E^p|$  – количество ребер соответствующего поуровневого графа  $G^p$ ,  $|E|$  – количество дуг графа  $G$  и  $L_p$  – суммарная длина дуг графа  $G$ .

**Теорема 8.** Общая сложность алгоритма связывания ребер для поуровневых графов составляет  $O(m^4 + nt)$ , где  $n$  – количество вершин, а  $t$  – количество ребер графа  $G^p$ .

Тестирование метода связывания ребер проводилось на реальных и синтетических безмасштабных графах. Использовались коллекции графов, сгенерированные в лаборатории AT&T и являющиеся де-факто стандартными для тестирования поуровневых упадок. Также рассматривались укладки графов зависимостей программных проектов, построенных в компании Microsoft. В целом метод связывания ребер оказался полезным на практике. Он строит значительно более простые и эстетически привлекательные изображения, чем классический метод Сугиямы, а его время выполнения сравнимо со временем выполнения других этапов алгоритма рисования поуровневых графов.

В главе 4 описываются модели рисования динамических графов. Для того чтобы наглядно подчеркнуть происходящие изменения в сети, используется модель *послойных* упадок графов с сохранением позиций вершин. Метод послойной визуализации состоит из следующих шагов.

- Рассматриваемый период времени разбивается на непересекающиеся подпериоды. Строится последовательность графов, моделирующая исходные данные за каждый подпериод.
- Для каждого построенного графа строится укладка на плоскости.
- Результирующая визуализация получается объединением всех упадок с соединением вершин, соответствующим одинаковым объектам в соседних графах (рис. 3(c)).



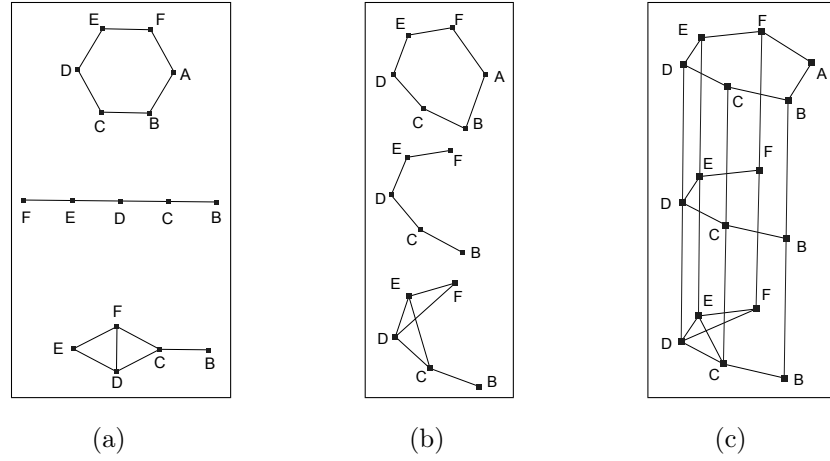


Рис. 3. Граф, состоящий из трех слоев. (a) Укладки без учета ментальной карты. (b) Сохранение ментальной карты. (c) Послойная визуализация.

В задаче динамической визуализации требуется нарисовать последовательность графов  $G_1, G_2, \dots, G_T$ , описывающих некоторые данные за последовательные промежутки времени. С каждой вершиной  $v \in G_i$  ассоциирована метка  $l_v$ . Будем считать, что  $l_v = l_u$  тогда и только тогда, когда вершины  $v$  и  $u$  соответствуют одному и тому же объекту исходных данных. При построении послойной визуализации требуется учитывать два конкурирующих критерия. С одной стороны, каждый слой должен быть нарисован с учетом общепринятых критериев эстетичности графа: малое количество пересечений ребер, симметрия, постоянная длина ребер и т.д. С другой, последовательность упадок должна сохранять ментальную карту [5]. Это означает, что графы нужно рисовать в унифицированном стиле: вершины с одинаковыми метками должны иметь близкие позиции.

Для решения поставленной задачи добавим в силовую и пружинную модели дополнительные силы:

- ментальные силы, притягивающие вершины с одинаковыми метками. Благодаря этому, исходные объекты будут стремиться занять одинаковые позиции во всех слоях результирующей послойной укладки, что соответствует сохранению ментальной карты.
- гравитационные силы, притягивающие все вершины к фиксированной точке. Без гравитации минимум энергии будет соответствовать укладке, в которой разные компоненты связности графа находятся на

бесконечном расстоянии, т.к. между вершинами из разных компонент связности действуют только силы отталкивания.

Итоговое выражение для энергии системы выглядит следующим образом:

$$U = U_{attr} + U_{repu} + U_{grav} + U_{mental}. \quad (2)$$

Остается открытым вопрос, как минимизировать  $U$ . Заметим, что на последовательность  $G_1, G_2, \dots, G_T$  графов можно смотреть, как на один большой граф, объединяющий в себе все вершины и ребра каждого графа последовательности. Граф-объединение  $G_s = (V_s, E_s)$  состоит из множества вершин  $V_s = \cup_{i=1}^T V^i \cup v_{grav}$  и ребер  $E_s = \cup_{i=1}^T E^i \cup \{(u, v) \subseteq V_s \times V_s : l_u = l_v\} \cup \{(u, v_{grav}), u \in \cup_{i=1}^T V_i\}$ . В этом смысле компоненты  $U_{grav}$  и  $U_{mental}$  – это части энергии притяжения. Поэтому для минимизации выражения 2 можно применять классические итеративные методы минимизации энергии для статических графов. В диссертации установлено (**теоремы 9 и 10**), что сложность алгоритма визуализации последовательности графов  $G_1, G_2, \dots, G_T$  равна суммарной сложности визуализаций каждого из графов  $G_1, G_2, \dots, G_T$ .

Описанный метод был применен для анализа сложных динамических сетей. В качестве входных данных нами были выбраны несколько общедоступных популярных социальных сетей, среди которых сеть контактов e-mail компании Enron и архив сообщений системы Twitter. Проведенные исследования подтверждают, что методы динамической визуализации помогают при решении прикладных задач анализа социальных отношений. В частности, нами были выявлены основные шаблоны общения пользователей, изучены изменения в группах общения, проведены исследования динамики интенсивности общения в сообществе.

**Пятая глава** посвящена проекту GraphVis3D – разработанному программному комплексу визуализации и анализа сложных сетей. В главе описываются основные возможности комплекса, рассказывается об его архитектуре. Приведено детальное сравнение с аналогичными программными системами, сформулированы основные недостатки существующих систем.

С технической точки зрения система визуализации GraphVis3D явля-

ется Java-библиотекой, реализованной на базе платформы Java3D, и поэтому является кросс-платформенной, что выгодно отличает ее от аналогов, ориентированных на конкретную операционную систему. Простой пользовательский интерфейс и широкий набор графических средств визуализации делают разработанную систему удобным инструментом анализа сложных данных. Отметим основные возможности комплекса.

- **Интерактивность.** Пользователь может редактировать элементы графа и тут же получать результат. Интерактивность относится и к алгоритмам укладки: пользователь может просматривать пошаговое выполнение алгоритма, управлять его параметрами в ходе выполнения.
- **Поддержка динамических графов.** Для визуализации таких графов используется послойный метод рисования. При этом пользователь может отслеживать динамику отдельных вершин и сообществ.
- **Расширяемая архитектура** позволяет легко добавлять новую функциональность в систему. От программиста, создающего свое собственное расширение, не требуется знания деталей внутреннего представления графов и логики работы с ними.
- **Богатый набор графических возможностей и пользовательского интерфейса** (поддержка стилей, операции Undo/Redo и т.д.).
- **Встроенная возможность автоматического анализа сетей** (фильтрация, статистика, аннотация данных) и укладок графов (автоматическое вычисление метрик эстетичности).

Система доступна для свободного скачивания и использования на ресурсе <http://code.google.com/p/graphvis>. Там же доступна подробная инструкция пользователю и документация.

В **заключении** приводятся основные выводы и результаты работы. Формулируются актуальные направления дальнейших исследований по данной тематике.

## СПИСОК ЛИТЕРАТУРЫ

- [1] *Albert R., Barabasi A.-L.* Statistical mechanics of complex networks // *Reviews of Modern Physics*. — 2002. — Vol. 74. — Pp. 47–97.
- [2] *Eades P.* A heuristic for graph drawing // *Congressus Numerantium*. — 1984. — Vol. 42. — Pp. 149–160.
- [3] *Fortunato S., Castellan C.* Community structure in graphs / Ed. by B. Meyers. — *Encyclopedia of Complexity and System Science*. 2008.
- [4] *Girvan M., Newman M. E. J.* Community structure in social and biological networks // *Proc. of the National Academy of Sciences of the USA*. — 1999. — Pp. 7821–7826.
- [5] Layout adjustment and the mental map / K. Misue, P. Eades, W. Lai, K. Sugiyama // *Journal of Visual Languages and Computing*. — 1995. — Vol. 6, no. 2. — Pp. 183–210.
- [6] Line crossing minimization on metro maps / M. A. Bekos, M. Kaufmann, K. Potika, A. Symvonis // *Proc. 15th Int. Symposium on Graph Drawing*. — 2008. — Pp. 231–242.
- [7] *Moreno J. L.* Application of the group method to classification // *New York: National Committee on Prisons and Prison Labor*. — 1932.
- [8] *Newman M. E. J., Girvan M.* Finding and evaluating community structure in networks // *Physical Review E*. — 2004. — Vol. 69. — P. 026113.
- [9] A short note on the history of graph drawing / E. Kruja, J. Marks, A. Blair, R. Waters // *Proc. 9th Int. Symposium on Graph Drawing*. — 2001. — Pp. 602–606.
- [10] *Sugiyama K., Tagawa S., Toda M.* Methods for visual understanding of hierarchical system structures // *IEEE Transactions on Systems, Man, and Cybernetics*. — 1981. — Vol. 11, no. 2. — Pp. 109–125.

## РАБОТЫ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ

**Статьи, опубликованные в ведущих рецензируемых научных журналах, определенных ВАК:**

- [11] Пупырев С.Н. *Визуализация структуры сообществ в графах* // Системы управления и информационные технологии. – Воронеж, 2009. № 2(36). С. 21–27.
- [12] Пупырев С.Н., Тихонов А.В. *Визуализация динамических графов для анализа сложных сетей* // Моделирование и анализ информационных систем. – Ярославль, 2010. № 17(1). С. 117–135.
- [13] Pupyrev S., Nachmanson L., Kaufmann M. *Improving layered graph layouts with edge bundling* // Proceedings of 18th International Symposium on Graph Drawing, Konstanz, Germany. – Springer-Verlag, 2010. № 6502. С. 147–158.

### **Другие публикации:**

- [14] Nachmanson L., Pupyrev S. *Visualizing a layered graph using edge bundling* // U.S. patent application MS#328544.01, 2010.
- [15] Пупырев С.Н. *Визуализация безмасштабных графов* // Труды 40-й всероссийской молодежной школы-конференции. – Екатеринбург, 2009. № 40. С. 374–378.
- [16] Пупырев С.Н. *Эффективное вычисление метрик эстетичности* // Известия Уральского государственного университета, серия Компьютерные науки и информационные технологии. – Екатеринбург, 2010. № 74. С. 171–179.
- [17] Пупырев С.Н. *Система визуализации графов GraphVis3D* // Труды 41-й всероссийской молодежной школы-конференции. – Екатеринбург, 2010. № 41. С. 501–507.
- [18] Пупырев С.Н., Пронченков А.Г. *Прогнозирование загруженности автомобильных дорог* // Труды конференции молодых ученых по информационному поиску. – Воронеж, 2010. № 4. С. 64–78.